



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,960	11/24/2003	Richard D. Dettinger	ROC920030277US1	5201
46797	7590	11/01/2006		EXAMINER
IBM CORPORATION, INTELLECTUAL PROPERTY LAW DEPT 917, BLDG. 006-1 3605 HIGHWAY 52 NORTH ROCHESTER, MN 55901-7829			DWIVEDI, MAHESH H	
			ART UNIT	PAPER NUMBER
				2168

DATE MAILED: 11/01/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	10/720,960	DETTINGER ET AL.
	Examiner Mahesh H. Dwivedi	Art Unit 2168

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1) Responsive to communication(s) filed on 18 August 2006.
 2a) This action is **FINAL**. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4) Claim(s) 10-12, 14-21 and 27-29 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 10-12, 14-21 and 27-29 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on 11/24/2003 is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) <input type="checkbox"/> Notice of References Cited (PTO-892)	4) <input type="checkbox"/> Interview Summary (PTO-413)
2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)	Paper No(s)/Mail Date: _____
3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date: _____	5) <input type="checkbox"/> Notice of Informal Patent Application
	6) <input type="checkbox"/> Other: _____

DETAILED ACTION

Information Disclosure Statement

1. The information disclosure statements (IDS) submitted on 10/27/2004 and 11/24/2003 have been received, entered into the record, and considered. The submission is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statements are being considered by the examiner.

Response to Amendment

2. Receipt of Applicant's Amendment, filed on 08/18/2006, is acknowledged. The amendment includes amending the specification, the cancellation of Claims 1-9, 13, and 22-26, and amending Claims 10, 14-15, 20, and 27.

Remarks

3. The examiner notes that there are several discrepancies between the amended set of claims and that of the originally filed set of claims on 11/24/2003. For example, in claim 20, the phrase "**based on the runtime metadata and metadata associated with functional modules**" is deleted. However, the phrase "**and metadata associated with functional modules**" was never in the original claim 20. The examiner states that amendments should be made to the originally filed claims.

Specification

4. The objections raised in the office action mailed on 05/18/2006 have been overcome by the applicant's amendments received on 08/18/2006.

Drawings

5. The objections raised in the office action mailed on 05/18/2006 have been overcome by the applicant's amendments received on 08/18/2006.

Claim Objections

6. Claim 20 is objected to because of the following informalities: The examiner suggests that applicants change "obtaining a list of functional modules accessible from within the application" to "obtaining a list of functional modules accessible from within the application;". Appropriate correction is required.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

8. Claims 10-12, 14-21, and 27-29 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Win et al.** (U.S. Patent 6,453,353) and in view of **Pazandak et al.** (U.S. Patent 7,027,975).

9. Regarding claim 10, **Win** teaches a method comprising:

- A) assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database (**Win**, Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- B) wherein the assigned metadata requirements specify conditions required for successful execution of the functional module (**Win**, Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- C) obtaining a list of functional modules that are accessible from within the application used during the query session (**Win**, Abstract, Column 6, lines 10-16, lines 41-65);

- D) identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements (Win, Abstract, Column 6, lines 10-16, lines 41-65); and
- E) providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute (Win, Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches "assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database" as "Roles determine what resources a User can access. Further, each role may require a set of information that is available in resources" (Column 5, lines 44-46) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches "wherein the assigned metadata requirements specify conditions required for successful execution of the functional module" as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches "obtaining a list of functional modules that are accessible from within the application used during the query session" as "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches "identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the

assigned metadata requirements" as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that the user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches "providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute" as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15).

Win does not explicitly teach:

- F) collecting runtime metadata relating to the query session; and
- G) wherein the metadata is collected after the composition of a query.

Pazandak, however, teaches "collecting runtime metadata relating to the query session collecting runtime metadata relating to the query session" as "LL interface descriptor (B), as defined in more detail below, which is sent to the Parser Farm 308 on the server below" (Column 12, lines 19-22) and "The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata" (Column 17, lines 16-20) and "wherein the metadata is collected after the composition of a query" as "LL interface descriptor (B), as defined in more detail below, which is sent to the Parser Farm 308 on the server below" (Column 12, lines 19-22) and "The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata" (Column 17, lines 16-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more

scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 11, **Win** does not explicitly teach a method comprising:

A) wherein the runtime metadata comprises attributes of fields involved in a query or query results.

Pazandak, however, teaches “**wherein the runtime metadata comprises attributes of fields involved in a query or query results**” as “Sentence Construction” (Column 14, line 4, Figures 7-9, 11), and “Choice items” (Column 15, line 2, Figures 7-9, 11).

The examiner notes that the various screenshots depicted in Figures 7-9, and 11 all depict several attributes associated with various fields of query construction.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak**’s would have allowed **Win**’s to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 12, **Win** does not explicitly teach a method comprising:

A) wherein the runtime metadata comprises content contained in query results.

Pazandak, however, teaches “**wherein the runtime metadata comprises content contained in query results**” as “execution results of a sample LL query in a tabular format” (Column 32, lines 48-49, Figure 18).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak**’s would have allowed **Win**’s to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 14, **Win** further teaches a method comprising:

A) wherein obtaining metadata associated with the functional module comprises examining a signature validation (Column 6, lines 1-3, Column 14, lines 34-43).

The examiner notes that **Win** teaches “**wherein obtaining metadata associated with the functional module comprises examining a signature validation**” as “users may log in either with a digital certificate or by opening a login page URL with a web browser and entering a name and password” (Column 6, lines 1-3).

Regarding claim 15, **Win** further teaches a method comprising:

A) wherein the metadata associated with at least one of the functional modules comprises at least one of: one or more input parameters required for successful execution of the functional module, one or more output parameters required for successful execution of the functional module, and a security credential required to execute the functional module (Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches “**wherein the metadata associated with at least one of the functional modules comprises at least one of one or more input parameters required for successful execution of the functional module; one or more output parameters required for successful execution of the functional module; and a credential of a user authorized to execute the functional module**” as “The Authentication Client Module and Access Menu Module authenticates a user by verifying the name and password with the Registry Server 108” (Column 6, lines 42-44).

Regarding claim 16, **Win** does not explicitly teach a method comprising:

A) wherein at least one of the functional modules analyzes query results.

Pazandak, however, teaches “**wherein at least one of the functional modules analyzes query results**” as “a sophisticated extension is a capability for cooperative response, where if the user’s query results in a request for which no items are returned (it is overly constrained), then the translation tree can be examined to examine the results” (Column 32, lines 65-67-Column 33, line 1), and “A further sophisticated extension is to automatically relax some criteria” (Column 33, lines 9-10)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 17, **Win** does not explicitly teach a method comprising:

- A) the runtime metadata comprises the names of fields in a result set; and
- B) the limited subset of functional modules comprises functional modules requiring data from fields in the result set as inputs.

Pazandak, however, teaches “**the runtime metadata comprises the names of fields in a result set**” as “Categorization-Group items into (possibly hierarchical) sets of categories, whereby the user selects a category and is then shown the list of subcategories or choice items” (Column 14, lines 64-67, Figures 7-9, 11), and “**the limited subset of functional modules comprises functional modules requiring data from fields in the result set as inputs**” as “Categorization-Group items into (possibly hierarchical) sets of categories, whereby the user selects a category and is then shown the list of subcategories or choice items” (Column 14, lines 64-67, Figures 7-9, 11).

The examiner notes that the subcategories depicted in Figure 8 are analogous to selecting various fields to further limit a query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 18, **Win** does not explicitly teach a method comprising:

- A) wherein the runtime metadata comprises information related to a query building session.

Pazandak, however, teaches “**wherein the runtime metadata comprises information related to a query building session**” as “Appearance during sentence construction” (Column 14, line 4, Figures 1-9, 11).

The examiner notes that the various interfaces depicted in Figures 7-9, and 11 show query construction.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak**’s would have allowed **Win**’s to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 19, **Win** does not explicitly teach a method comprising:

- A) wherein the information related to the query building session comprises a specified focus of the query; and
- B) identifying a limited subset of the functional modules that will successfully execute comprises identifying functional modules associated with the specified focus.

Pazandak, however, teaches “**wherein the information related to the query building session comprises a specified focus of the query**” as “Categorization-Group items into (possibly hierarchical) sets of categories, whereby the user selects a category and is then shown the list of subcategories or choice items” (Column 14, lines 64-67, Figures 7-9, 11), and “**identifying a limited subset of the functional modules that will successfully execute comprises identifying functional modules associated with the specified focus**” as “Categorization-Group items into (possibly hierarchical) sets of categories, whereby the user selects a category and is then shown the list of subcategories or choice items” (Column 14, lines 64-67, Figures 7-9, 11).

The examiner notes that it is common knowledge that in a hierarchical category organization, selected different categories yields different subcategories. The examiner further notes that a category is analogous to a “**focus**”.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching

Pazandak's would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 20, **Win** teaches a method and computer readable medium comprising:

- A) assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database (**Win**, Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- B) wherein the assigned metadata requirements specify conditions required for successful execution of the functional module (**Win**, Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- C) obtaining a list of functional modules accessible from within the application (**Win**, Abstract, Column 6, lines 10-16, lines 41-65);
- D) identifying a limited subset of the functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements (**Win**, Abstract, Column 6, lines 10-16, lines 41-65); and
- E) providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute (**Win**, Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches "assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database" as "Roles determine what resources a User can access. Further, each role may require a set of information that is available in resources" (Column 5, lines 44-46) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that the user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches "wherein the assigned metadata requirements specify conditions

required for successful execution of the functional module" as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches "**obtaining a list of functional modules accessible from within the application**" as "a list of authorized resources" (Column 6, lines 13-14) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches "**identifying a limited subset of the functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements**" as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches "**providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute**" as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15).

Win does not explicitly teach:

F) collecting runtime metadata relating to a query session.

Pazandak, however, teaches "**collecting runtime metadata relating to a query session**" as "LL interface descriptor (B), as defined in more detail below, which

is sent to the Parser Farm 308 on the server below" (Column 12, lines 19-22) and "The Parser 310 can also send the set or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata" (Column 17, lines 16-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 21, **Win** does not explicitly teach a computer readable medium comprising:

- A) wherein the application is a query building application.

Pazandak, however, teaches "**wherein the application is a query building application**" as "Appearance during sentence construction" (Column 14, line 4, Figures 1-9, 11).

The examiner notes that the various interfaces depicted in Figures 7-9, and 11 show query construction.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 27, **Win** teaches a data processing system comprising:

- A) a data repository (**Win**, Column 5, lines 13-15);
- B) a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules (**Win**, Abstract, Column 6, lines 10-16, lines 41-65);
- C) an application from which the functional modules are accessible (**Win**, Abstract, Column 6, lines 10-16, lines 41-65);

D) present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules (Win, Abstract, Column 5, lines 66-67-Column 6, lines 1-16).

The examiner notes that Win teaches “**a data repository**” as “The system 2 enables organizations to register information sources or Resources and register Users of the information in a central repository” (Column 5, lines 13-15). The examiner further notes that Win teaches “**a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules**” as “a list of authorized resources” (Column 6, lines 13-14). The examiner further notes that Win teaches “**an application from which the functional modules are accessible**” as “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14). The examiner further notes that Win teaches “**present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules**” as “the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource” (Abstract), “If the name and password are correct, the Authentication Client Module reads the user’s roles from the Registry server” (Column 6, lines 44-46), and “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14).

Win does not explicitly teach:

E) wherein the application is configured to collect runtime metadata after the composition of a query.

Pazandak, however, teaches “**wherein the application is configured to collect runtime metadata after the composition of a query**” as “LL interface descriptor (B), as defined in more detail below, which is sent to the Parser Farm 308 on the server below” (Column 12, lines 19-22) and “The Parser 310 can also send the set

or a subset of the Interface Descriptor 306, e.g., LL Parser ID, transaction ID, or other metadata" (Column 17, lines 16-20).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 28, **Win** does not explicitly teach a data processing system comprising:

A) wherein the data repository comprises XML data structures used to store runtime metadata.

Pazandak, however, teaches "**wherein the data repository comprises XML data structures used to store runtime metadata**" as "The LL Client code can take any of several forms: a scripted document using scripting language such as Javascript and/or dynamic HTML an otherwise encoded document using a format such as XML, or an application-specific encoding an HTML document" (Column 16, line 54-64) and "Properties can be encoded in various ways; for instance in XML" (Column 24, lines 62-63).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Pazandak's** would have allowed **Win's** to provide a method to implement more scalable and easier-to-use thin interfaces for query construction and execution, as noted by **Pazandak** (Column 1, lines 60-67).

Regarding claim 29, **Win** further teaches a data processing system comprising:

A) wherein the data repository comprises relational database tables used to store runtime metadata (Column 5, lines 13-15, Column 7, lines 1-6).

The examiner notes that **Win** teaches "**wherein the data repository comprises relational database tables used to store runtime metadata**" as "The Registry

Repository is structured as a database. For example, the Registry Repository may be an SQL Server relational database management system, the Oracle 7® database, etc.” (Column 7, lines 1-6). The examiner further notes that it is common knowledge that relational databases store data in tables.

Response to Arguments

10. Applicant's arguments filed on 08/18/2006 have been fully considered but they are not persuasive.

Applicant goes on to argue on page 10, that “**Win does not disclose identifying a subset of functional modules base don a comparison of collected runtime metadata with metadata requirements that specify conditions required for successful execution of the functional modules, as specified in independent claims 20 and 27**”. However, the examiner wishes to point to Column 6 of **Win** and refer to the second paragraph which states “a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access” (Column 6, lines 12-15). The examiner further wishes to point to Column 6 of **Win** and refer to the seventh paragraph which states “When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource” (Column 6, lines 58-64). The examiner further wishes to state that the **Win**’s method clearly compares received metadata (see user inputted metadata) with required metadata (see stored passwords and roles) to output an appropriate set of resources via an HTML interface, based on the user’s role.

Applicant goes on to argue on page 10, that “**Applicants submit, however, that Win does not teach limiting functional modules (resources) that will successfully execute based on a comparison of collected runtime metadata and metadata requirements associated with the functional modules**”. However, the examiner wishes to point to Column 6 of **Win** and refer to the second paragraph which states “a Personalized Menu is an HTML page containing a list of authorized resources. The

Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15). The examiner further wishes to point to Column 6 of **Win** and refer to the seventh paragraph which states "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further wishes to state that the **Win's** method clearly compares received metadata (see user inputted metadata) with required metadata (see stored passwords and roles) to output an appropriate set of resources via an HTML interface, based on the user's role. The examiner further wishes to state that the resources of **Win** will only execute (i.e. appear on an HTML interface) if the User's has an acceptable role or security level.

Applicant goes on to argue on page 10, that "**In fact, Win is silent as to any determination of whether a functional module (resource) presented to a user will successfully execute, but rather is concerned only with a user's authentication**". However, the examiner wishes to point to Column 6 of **Win** and refer to the second paragraph which states "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15). The examiner further wishes to point to Column 6 of **Win** and refer to the seventh paragraph which states "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further wishes to state that the **Win's** method clearly compares received metadata (see user inputted metadata) with required metadata (see stored passwords and roles) to output an appropriate set of resources via an HTML interface, based on the user's role. The examiner further wishes to state that the resources of **Win** will only execute (i.e. appear on an HTML interface) if the User's has an acceptable role or security level.

Applicant goes on to argue on page 11, that “**the references would fail to teach identifying a subset of functional modules base don a comparison of collected runtime metadata with metadata requirements that specify conditions required for successful execution of the functional modules, as recited in independent claim 10**”. However, the examiner wishes to point to Column 6 of **Win** and refer to the second paragraph which states “a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access” (Column 6, lines 12-15). The examiner further wishes to point to Column 6 of **Win** and refer to the seventh paragraph which states “When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource” (Column 6, lines 58-64). The examiner further wishes to state that the **Win’s** method clearly compares received metadata (see user inputted metadata) with required metadata (see stored passwords and roles) to output an appropriate set of resources via an HTML interface, based on the user’s role.

Applicant goes on to argue on page 11, that “**there is no teaching of limiting a set of functional modules presented to a user to those that will successfully executed, as determined by a comparison of the collected runtime metadata and a set of associated metadata requirements**”. However, the examiner wishes to point to Column 6 of **Win** and refer to the second paragraph which states “a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access” (Column 6, lines 12-15). The examiner further wishes to point to Column 6 of **Win** and refer to the seventh paragraph which states “When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource” (Column 6, lines 58-64). The examiner further wishes to state that the **Win’s** method clearly

compares received metadata (see user inputted metadata) with required metadata (see stored passwords and roles) to output an appropriate set of resources via an HTML interface, based on the user's role.

Conclusion

11. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. PGPUB 2002/0083075 issued to **Brummel et al.** on 27 June 2002. The subject matter disclosed therein is pertinent to that of claims 1-29 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 6,757,898 issued to **Ilzen et al.** on 29 June 2004. The subject matter disclosed therein is pertinent to that of claims 1-29 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2003/0140043 issued to **Hotchkiss et al.** on 24 July 2003. The subject matter disclosed therein is pertinent to that of claims 1-29 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 20040249674 issued to **Eisenberg** on 09 December 2004. The subject matter disclosed therein is pertinent to that of claims 1-29 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2003/0229623 issued to **Chang et al.** on 11 December 2003. The subject matter disclosed therein is pertinent to that of claims 1-29 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2002/0091836 issued to **Moetelli** on 11 July 2002. The subject matter disclosed therein is pertinent to that of claims 1-29 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 6,430,556 issued to **Goldberg et al.** on 06 August 2002. The subject matter disclosed therein is pertinent to that of claims 1-29 (e.g., methods to use plug-ins in an interface for role-based users).

12. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Contact Information

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi

Patent Examiner

Art Unit 2168


October 25, 2006


Leslie Wong


TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100